Università degli Studi di Palermo

dipartimento di ingegneria unipa

Introduzione al Modulo

CALCOLATORI ELETTRONICI – FONDAMENTI DI PROGRAMMAZIONE

a.a. 2023/2024

Prof. Roberto Pirrone

CHILAB COMPUTER HUMAN INTERACTION LABORATORY Deep Learning and NLP in Healthcare

Il Docente

- Roberto Pirrone
  - Studio: Edificio 6, terzo piano, stanza 3025
  - Email: roberto.pirrone@unipa.it,
    roberto.pirrone@community.unipa.it (Google)
  - Telefono studio: 091238.62625, laboratorio: .62643
  - Ricevimento: ogni mercoledì dalle 11:30 alle 13 presso il proprio studio

Il Laboratorio

- Laboratorio di Interazione Uomo-Macchina
  - Edificio 6, terzo piano, a sx dalle scale
  - Email: chilab@unipa.it
  - Telefono: 091238.62643

CHILAB
COMPUTER HUMAN INTERACTION LABORATORY
Deep Learning and NLP in Healthcare

Università degli Studi di Palermo
dipartimento di ingegneria unipa

E adesso ….

Cosa vi aspettate da questo corso?

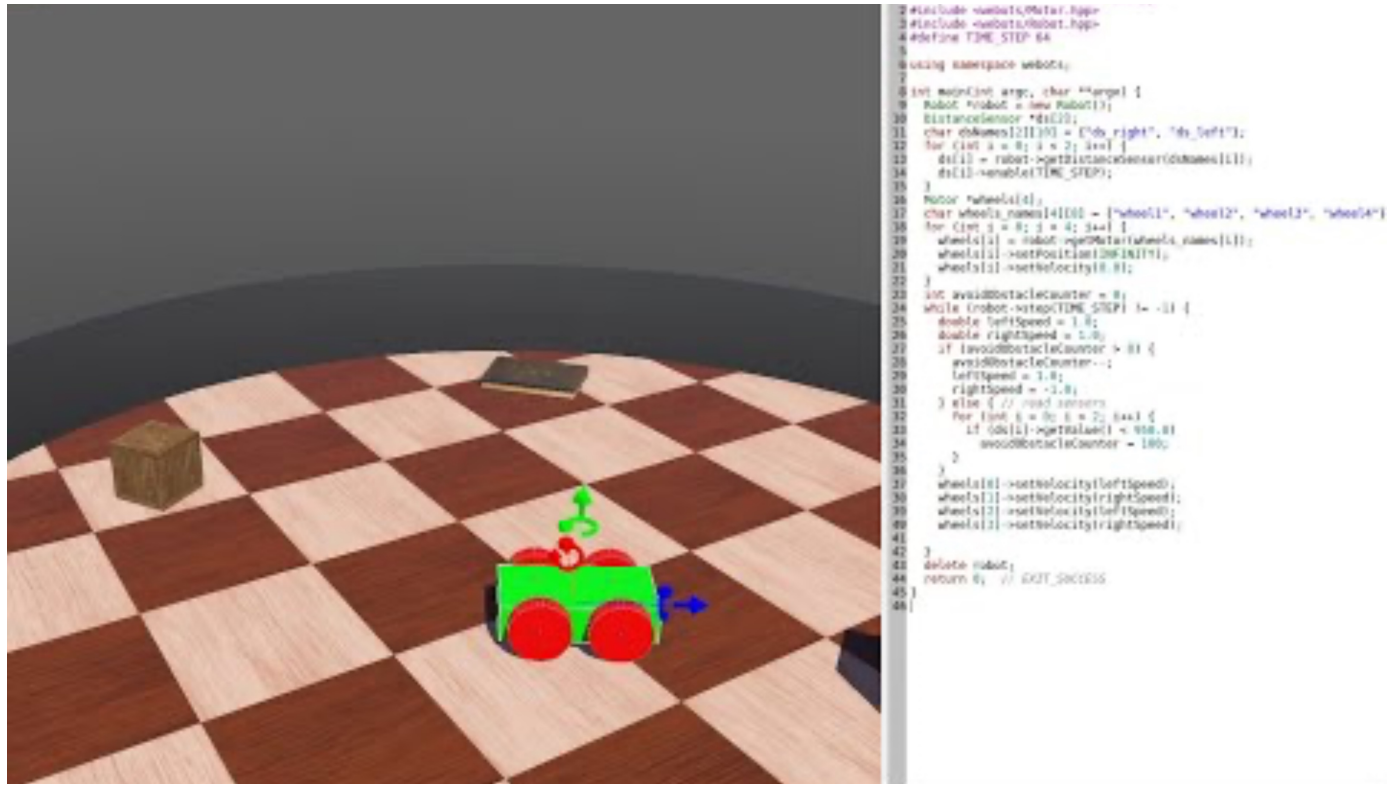
Cosa pensate che sia «Fondamenti di Programmazione»?

Cosa non è «Fondamenti di Programmazione»

- Il modulo di «Fondamenti di Programmazione» *non è*:
  - Un corso programmazione dei Robot (è ancora troppo presto …)
  - Un tutorial del linguaggio C (anche se è il linguaggio che impareremo)
  - Un corso avanzato sul software per i calcolatori
    - Sistemi Operativi, Reti, Programmazione e così via hanno bisogno di intere materie dedicate

Cosa non è «Fondamenti di Programmazione»

- *Non è possibile* programmare subito i robot se non si impara a *programmare*



```
#include <webots/Motor.hpp>
#include <webots/Robot.hpp>
#define TIME_STEP 64

using namespace webots;

int main(int argc, char **argv) {
  Robot *robot = new Robot();
  DistanceSensor *ds[2];
  char dsNames[2][10] = {"ds_right", "ds_left"};
  for (int i = 0; i < 2; i++) {
    ds[i] = robot->getDistanceSensor(dsNames[i]);
    ds[i]->enable(TIME_STEP);
  }
  Motor *wheels[4];
  char wheels_names[4][30] = {"wheel1", "wheel2", "wheel3", "wheel4"};
  for (int i = 0; i < 4; i++) {
    wheels[i] = robot->getMotor(wheels_names[i]);
    wheels[i]->setPosition(INFINITY);
    wheels[i]->setVelocity(0.0);
  }
  int avoidObstacleCounter = 0;
  while (robot->step(TIME_STEP) != -1) {
    double leftSpeed = 1.0;
    double rightSpeed = 1.0;
    if (avoidObstacleCounter > 0) {
      avoidObstacleCounter--;
      leftSpeed = 1.0;
      rightSpeed = -1.0;
    } else { // read sensors
      for (int i = 0; i < 2; i++) {
        if (ds[i]->getValue() < 950.0)
          avoidObstacleCounter = 100;
      }
    }
    wheels[0]->setVelocity(leftSpeed);
    wheels[1]->setVelocity(rightSpeed);
    wheels[2]->setVelocity(leftSpeed);
    wheels[3]->setVelocity(rightSpeed);

  }
  delete robot;
  return 0;  // EXIT_SUCCESS
}
```

Cosa è «Fondamenti di Programmazione»

- Il modulo di «Fondamenti di Programmazione» introduce i concetti di base della Programmazione Strutturata usando il linguaggio C

- Un buon programmatore deve conoscere:
  - Il funzionamento dell'hardware di riferimento (modulo precedente)
  - Le tecniche di programmazione (i fondamenti in questo modulo)

Cosa è «Fondamenti di Programmazione»

Algoritmi

+

Strutture Dati

=

PROGRAMMI

Niklaus Wirth

Inventore del Linguaggio PASCAL

Cosa è «Fondamenti di Programmazione»

- Algoritmo:

Un insieme ben ordinato di operazioni non ambigue ed effettivamente calcolabili che, eseguito, produce un risultato e termina in una quantità finita di tempo.

- Insieme ben ordinato
- Operazioni non ambigue e calcolabili
- Produce un risultato
- Termina in una quantità finita di tempo

Cosa è «Fondamenti di Programmazione»

- Struttura di dati:

Un'entità usata per organizzare un insieme di dati all'interno della memoria del computer, ed eventualmente per memorizzarli in una memoria di massa.

  - Metodo di organizzazione dati
  - Prescinde da ciò che è effettivamente contenuto
  - Ogni linguaggio di programmazione ne offre diverse
  - E' legata all'algoritmo e ne condiziona l'*efficienza*

Il Syllabus

- Le informazioni complete sugli obiettivi didattici del corso, il programma delle lezioni e i libri di testo si trovano nella *Scheda di Trasparenza*

Il Syllabus

- Testi consigliati
  - Jeri R. Anly – Elliot B. *Koffman*, Problem solving e programmazione in C, Apogeo, ISBN-10: 8838786410 (prezzo orientativo € 42,00)
  - J. Glenn *Brookshear* - Stephen G. Kochan, Fondamenti di informatica e programmazione in C, Pearson, ISBN-10: 8865183691 (prezzo orientativo € 34,00)

Il Syllabus

ORE	Lezioni Frontali	Testo rif.
1	Introduzione al Corso.	Slide docente
1	Elaborazione dei dati, architettura dei computer, sistemi operativi, reti di computer. Linguaggi di programmazione, prospettiva storica. Traduzione dei programmi. Indipendenza dalla macchina	Estratti dal Koffman Intro. e cap. 1 e dal Brookshear Intro.
2	Concetto di algoritmo, rappresentazione degli algoritmi, pseudocodice e diagrammi di flusso. Concetti della programmazione tradizionale. Compilazione del primo programma. Esecuzione del primo programma.	Koffman cap. 2
2	Introduzione al linguaggio C. Variabili, tipi di dati ed espressioni aritmetiche. Differenza tra variabili e costanti. Assegnamento. Operatori. Priorità tra gli operatori.	Koffman cap. 2
2	Concetto di sottoprogramma. Introduzione alle funzioni C. Valori di ritorno delle funzioni. Funzioni con parametri e senza parametri. Progettazione modulare del software: tecniche top-down e diagrammi di struttura. Comprensione e simulazione di algoritmi/programmi.	Koffman cap. 3

Il Syllabus

ORE	Lezioni Frontali	Testo rif.
2	Strutture di selezione (if-else) e di selezione tra più alternative (switch-case). Diagrammi di flusso per le strutture di selezione. Comprensione e simulazione di algoritmi/programmi.	Koffman cap. 4
2	Programmi iterativi in linguaggio C. Strutture iterative (for, while e do-while). Diagrammi di flusso per le strutture di iterative. Annidamento delle strutture. Comprensione e simulazione di algoritmi/programmi.	Koffman cap. 5
2	Algoritmi iterativi. Efficienza e correttezza di un algoritmo. Introduzione all'analisi computazionale di un algoritmo: complessità di tempo e di memoria.	Slide docente
3	Approfondimento sulle funzioni. Variabili locali. Visibilità e scope di una variabile. Variabili globali, automatiche e statiche. Concetto di puntatore e suo uso nei parametri di funzione.	Estratti dal Koffman cap. 6
2	Concetto di ricorsione. Gestione della memoria di una funzione ricorsiva. Implementazione in linguaggio C.	Estratti dal Koffman cap. 9

Il Syllabus

ORE	Lezioni Frontali	Testo rif.
3	Concetto di struttura di dati. Gli array: dichiarazione e indicizzazione. Array multidimensionali. Tipo enumerativo e array con indice di tipo enumerativo. Cicli e array. Semplici strutture dati con array: le pile.	Koffman cap. 7
2	Stringhe di caratteri. Utilizzo delle stringhe in C: array di caratteri, stringhe di caratteri di lunghezza variabile, sequenze di escape.	Koffman cap. 8
4	Approfondimento sui puntatori: puntatori e array. Aritmetica dei puntatori. Strutture e unioni. Implementazione di strutture e tipi personalizzati. Implementazione di strutture dati elementari: liste e code. Operazioni di ricerca e di ordinamento in collezioni di dati.	Estratti dal Koffman cap. 13
2	Gestione di input e output nel linguaggio C. Gestione di file.	Estratti dal Koffman cap. 11
2	Direttive di preprocessore. Inclusione di librerie personali.	Estratti dal Koffman cap. 12

Il Syllabus

ORE	Esercitazioni
4	Progettazione di semplici algoritmi attraverso diagrammi di flusso. Passaggio da diagrammi di flusso a codice C. Primi programmi in C: compilazione, linking ed esecuzione. Implementazione di programmi per la manipolazione di dati numerici.
2	Progettazione di algoritmi che prevedono la selezione e strutture iterative attraverso i diagrammi di flusso. Implementazione corrispondente in linguaggio C.
3	Esercizi sulla progettazione di software modulare attraverso un uso corretto delle funzioni. Progettazione e implementazione di algoritmi che utilizzano la ricorsione.
6	Progettazione e implementazione di algoritmi che utilizzano stringhe di caratteri, vettori e matrici. Utilizzo dei puntatori. Esercizi sull'implementazione di liste, code e pile.
3	Implementazione di software per l'I/O da file.
3	Preparazione alla prova scritta.

- Le esercitazioni saranno svolte in aula, usando Google Colab per comodità
- I codici risultanti verranno condivisi dal docente nel repository del corso

Il materiale didattico

- Le slide da sole *non sono* materiale didattico: esse sono a compendio dei libri di testo, della spiegazione orale del docente e degli *appunti* presi dallo studente

- *Suggerimento*: stampate le slide prima della lezione e annotatele con i vostri appunti

Il materiale didattico

- Repository GitHub del corso
  - Contiene:
    - I file pdf di tutte le slide (incluse queste)
    - I codici delle esercitazioni
    - I dati utilizzati nelle esercitazioni

Gli esami

- Compito scritto al calcolatore di programmazione C
  - Prova in itinere alla fine del corso (intorno ai primi di giugno)
  - Chi non la supera dovrà fare l'esame che è un compito unico sui due moduli che consta appunto di due sezioni
  - Il voto finale dell'intero corso è la media dei voti riportati nelle due prove in itinere dei due moduli ovvero deriva dal compito scritto
  - Eventuali domande orali saranno fatte a insindacabile giudizio della commissione per chiarire meglio eventuali parti dello scritto.